

**WHAT IS CLAIMED IS:**

1           1.    A computer-implemented method of searching a  
2    database for a prefix representing a destination address  
3    comprising:

4           loading two trees of tables, each tree of tables having a  
5    large table at a root branching to small tables; and

6           traversing the two tables of trees in parallel to find a  
7    match of an entry to the prefix.

1           2.    The computer-implemented of claim 1 wherein an entry  
2    comprises:

3           a router pointer representing the destination address;  
4    and

5           a pointer to a next small table.

1           3.    The computer-implemented method of 1 wherein the  
2    small tables comprise:

3           prefix match fields for indexed table entries;

4           a population count of pointers; and

5           hidden prefix entries that hold shorter prefix route  
6    entry pointers.

1           4.    The computer-implemented method claim 1 further  
2    comprising reporting a non-match if the prefix does not match  
3    an entry.

1           5.    The computer-implemented method of claim 1 wherein a  
2    first large table is a single 64k entry table that is indexed  
3    by bits 31:16 of an internet protocol (IP) address.

1           6.    The computer-implemented method of claim 1 wherein a  
2   second large table is a single 256 entry table that is indexed  
3   by bits 31:24 of an internet protocol (IP) address.

1           7.    The computer-implemented method of claim 5 wherein  
2   the small tables are dynamically allocated and comprise:

3           a tree with each node representing 4 bits of addresses  
4   covering an extension of 1-4 bits of a prefix entry from a  
5   previous tree.

1           8.    The computer-implemented method of claim 6 wherein  
2   the small tables are dynamically allocated and comprise:

3           a tree with each node representing 4 bits of addresses  
4   covering an extension of 1-4 bits of a prefix entry from a  
5   previous tree.

1           9.    A computer storage device storing a data structure  
2   for managing prefix representing internet protocol (IP)  
3   destination addresses, the data structure comprising:

4           two trees of tables, each tree of tables comprising:  
5           a trie block, the trie block including a route pointer  
6   and a trie pointer;

7           a trie information structure, the trie information  
8   structure including masks and route entry pointers.

1           10.   A computer-implemented method of searching a  
2   collection of data comprising:

3           searching a first table of trees and a second table of  
4   trees for a received search term, each of the trees of the  
5   first table and the second table containing a trie element and

6 a trie pointer, for a match of the search term with a trie  
7 element;

8 determining whether a trie pointer is non-null when the  
9 trie element matches the search term;

10 comparing a trie element in the tree of the first table  
11 containing the null pointer with a trie element in the tree of  
12 the second table containing the null pointer;

13 reporting a match if the search term matches the trie  
14 element in the first table of trees; and

15 reporting a match of the search term matches the trie  
16 element in the second table of trees.

1 11. The computer-implemented method of claim 10 wherein  
2 the search term is a destination address.

1 12. The computer-implemented method of claim 11 wherein  
2 the destination address is a prefix.

1 13. A computer program product, disposed on a computer  
2 readable medium, for searching a database for a prefix  
3 representing a destination address, the program comprising  
4 instructions for causing a computer to:

5 load two trees of tables, each tree of tables having a  
6 large table at a root branching to small tables; and

7 traverse the two tables of trees in parallel to find a  
8 match of an entry to the prefix.

1 14. The computer program of claim 13 wherein an entry  
2 comprises:

3 a router pointer representing the destination address;  
4 and

5 a pointer to a next small table.

1 15. The computer program of claim 13 wherein the small  
2 tables comprise:

3 prefix match fields for indexed table entries;

4 a population count of pointers; and

5 hidden prefix entries that hold shorter prefix route  
6 entry pointers.

1 16. The computer program claim 13 further comprising  
2 instructions for causing the computer to report a non-match if  
3 the prefix does not match an entry.

1 17. The computer program of claim 13 wherein a first  
2 large table is a single 64k entry table that is indexed by  
3 bits 31:16 of an internet protocol (IP) address.

1 18. The computer program of claim 13 wherein a second  
2 large table is a single 256 entry table that is indexed by  
3 bits 31:24 of an internet protocol (IP) address.

1 19. The computer program of claim 17 wherein the small  
2 tables are dynamically allocated and comprise a tree with each  
3 node representing 4 bits of addresses covering an extension of  
4 1-4 bits of a prefix entry from a previous tree.

1 20. The computer program of claim 18 wherein the small  
2 tables are dynamically allocated and comprise a tree with each  
3 node representing 4 bits of addresses covering an extension of  
4 1-4 bits of a prefix entry from a previous tree.

1 21. The computer-implemented method of claim 3 further  
2 comprising:

3 adding entries; and

4 deleting entries.

1           22. The computer-implemented method of claim 21 wherein  
2 deleting entries comprises:

3           removing corresponding trie entries;

4           decrementing the population counter;

5           determining an entry next longest prefix; and

6           inserting the next longest prefix in the trie.

006200-4580360